

一篇神文让你“一夜封神”Mycat中间件(最详细讲解)

原创

普通网友 于 2021-10-27 20:21:11 发布 35 收藏

文章标签: [java](#) [后端](#) [架构](#)

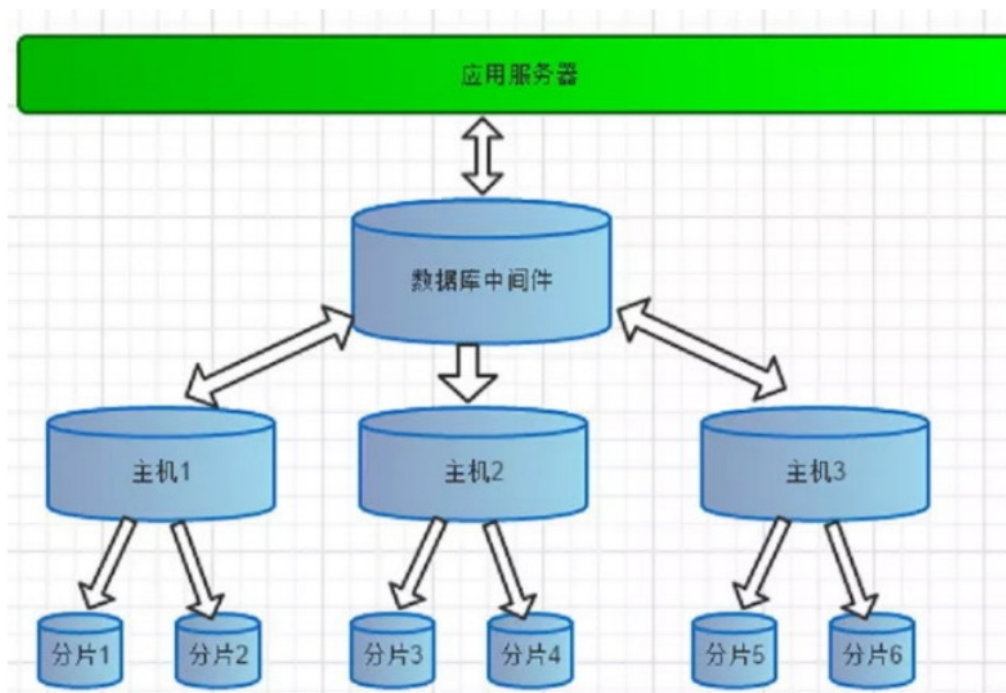
版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/m0_63551359/article/details/121001103

版权

Mycat简介

Mycat的目的是打造真正的分布式数据库中间件, 如图13-1所示。用户可以把Mycat看作是一个数据库代理, 用MySQL客户端工具和命令行访问, 而其后端可以用MySQL原生协议与多个MySQL服务器通信, 也可以用JDBC协议与大多数主流数据库服务器通信。可以像使用MySQL一样使用Mycat, 对于开发人员来说根本感觉不到Mycat的存在。



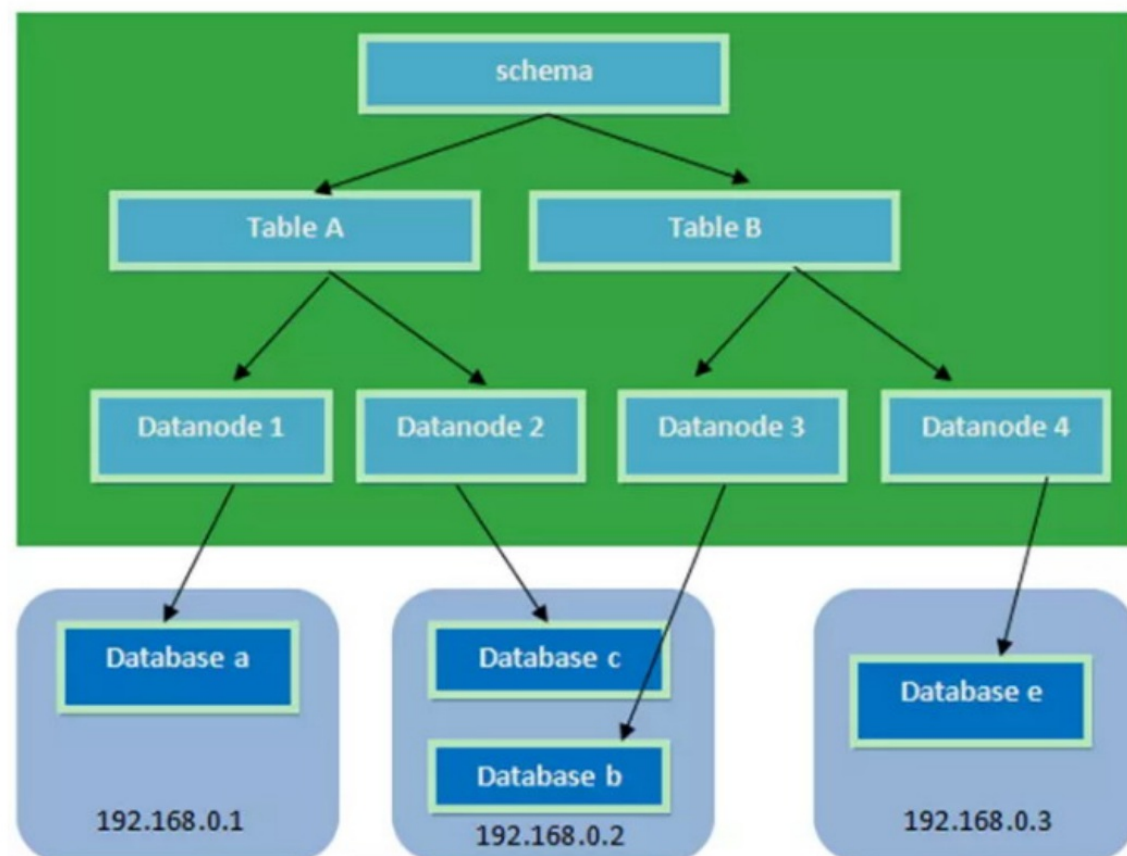
Mycat的核心功能是分表分库, 即将一个大表水平分割为N个小表, 存储在后端多个数据库(主机)里, 以达到分散单台设备负载的效果。应用程序就像连接普通的MySQL数据库一样地去连接它, SQL查询、操作等一模一样; 而Mycat把数据库复杂的架构以及背后复杂的分表分库的逻辑全部透明化了。Mycat中间件连接多个MySQL数据库, 多个数据库之间还可以做主从同步, 这一切的一切对前端应用是完全透明的, 不用调整前台逻辑, 只要连接到Mycat即可。这样一来, 对前端业务系统来说, 业务代码无须过多调整, 可以大幅降低开发难度, 提升开发速度。

Mycat可实现数据库的读写分离, 在后端的主从复制数据库集群中, 通过Mycat配置将前端的写操作路由到主数据库中, 将读操作路由到从数据库上。Mycat可以实现读写分离下的读操作负载均衡, 将大量的读操作均衡到不同的从库上, 主要出现在一主多从的情形下。

Mycat可实现数据库的高可用, 在数据库主节点可用的情况下, 配置一台可写从节点, 这两个节点都配置在Mycat中, 当主节点宕机时, Mycat会自动将写操作路由到备用节点上, 轻松实现热备份。

Mycat核心概念

Mycat的核心概念如图



(1) 逻辑库 (schema)

Mycat作为一个中间件，实现MySQL协议，对前端应用连接来说就是一个数据库，无须让开发人员知道中间件存在，所以数据库中间件可以被当作一个或多个数据库集群构成的逻辑库。

(2) 逻辑表 (table)

有逻辑库，就会有逻辑表。在分布式数据库中，对应用来说，读写数据的表就是逻辑表。逻辑表可以数据切分后分布在一个或多个分片库中，也可以不做数据切分，不分片，只由一个表构成。

(3) 分片表

分片表是指那些原有的拥有很多数据、需要切分到多个数据库的表。每个分片都有一部分数据，所有分片构成了完整的数据。

(4) 非分片表

一个数据库中并不是所有的表都很大，某些表是可以不进行切分的。非分片表是相对分片表来说的，就是那些不需要进行数据切分的表。

(5) ER表

关系型数据库是基于实体关系模型 (Entity-Relationship Model) 之上的，描述了真实世界中的事物与关系。Mycat中的ER表就来源于此。根据这一思路，提出了基于E-R关系的数据分片策略，子表的记录与所关联的父表记录存放在同一个数据分片上，即子表依赖于父表，通过表分组 (Table Group) 保证数据join关联查询不会跨库操作。表分组 (Table Group) 是解决跨分片数据join关联查询的一种很好的方法决定方法，也是数据切分规划很重要的一条原则。

(6) 全局表

在业务系统中，往往存在大量类似字典表的表，基本上很少变动。字典表的特性是：变动不频繁；数据量总体变化不大；数据规模不大，很少有超过数十万条记录的。当业务表因为数据量规模大而进行分片以后，业务表与这些附属的字典表之间的关联就成了比较棘手的问题，所以Mycat中通过数据冗余来解决这类表的join关联查询，即所有的分片都有一份数据的备份，所有将字典表或者符合字典表特性的一些表定义为全局表。数据冗余是解决跨分片数据join关联查询的一种很好的解决方法，也是数据切分规划的另一种外一条重要原则。

（7）分片节点（dataNode）

数据切分后，一个大表被分到不同的分片数据库上面，每个表分片所在的数据库就是分片节点。

（8）节点主机（dataHost）

数据切分后，每个分片节点不一定会独占一台机器，同一机器上面可以有多个分片数据库，这样一个或多个分片节点所在的机器就是节点主机。为了规避单节点主机并发次数限制，尽量将读写压力高的分片节点均衡地放在不同的节点主机上。

（9）分片规则（rule）

前面讲了数据切分，一个大表会被分成若干个分片表。按照某种业务规则把数据分到某个分片的规则就是分片规则。数据切分时选择合适的分片规则非常重要，这将极大地避免后续数据处理的难度。

（10）全局序列号（sequence）

数据切分后，原有的关系数据库中的主键约束在分布式条件下将无法使用，因此需要引入外部机制保护证明数据唯一性标识，这种保证全局性的数据唯一标识的机制就是全局序列号。